

THREAD DISPATCHER FOR MULTI-THREADED COMMUNICATION LIBRARY

Abstract of the Disclosure:

Method, computer program product, and apparatus for efficiently dispatching threads in a multi-threaded communication library which become runnable by completion of an event. Each thread has a thread-specific structure containing a "ready flag" and a POSIX thread condition variable unique to that thread. Each message is assigned a "handle". When a thread waits for a message, thread-specific structure is attached to the message handle being waited on, and the thread is enqueued, waiting for its condition variable to be signaled. When a message completes, the message matching logic sets the ready flag to READY, and causes the queue to be examined. The queue manager scans the queue of waiting threads, and sends a thread awakening condition signal to one of the threads with its ready flag set to READY. The queue manager can implement any desired policy, including First-In-First-Out (FIFO), Last-In-First-Out (LIFO), or some other thread priority scheduling policy. This ensures that the thread which is awakened has the highest priority message to be processed, and enhances the efficiency of message delivery. The priority of the message to be processed is computed based on the overall state of the communication subsystem.